Radiology: Artificial Intelligence

EDITORIAL

Magician's Corner: 8: How to Connect an Artificial Intelligence Tool to PACS

Bradley J. Erickson, MD, PhD • *Felipe Kitamura, MD, MSc*

From the Department of Radiology, Mayo Clinic, 200 First St SW, Rochester, MN 55905 (B.J.E.); and Department of Diagnostic Imaging, Universidade Federal de São Paulo, São Paulo, Brazil (F.K.). Received May 18, 2020; revision requested June 16; revision received July 6; accepted July 8. **Address correspondence to** B.J.E. (e-mail: *bje@mayo.edu*).

Conflicts of interest are listed at the end of this article.

Radiology: Artificial Intelligence 2021; 3(1):e200105 • https://doi.org/10.1148/ryai.2021200105 • Content code: AI • ©RSNA, 2021

> *"We are like islands in the sea, separate on the surface but connected in the deep."*
>
> –William James

Previous articles in this series have discussed how to process and access image data, how to train a deep learning tool to classify images, how to improve the appearance of images, how to segment the components of an image, and how to see what the algorithm is using to make its decisions. In this article, we provide some tools to help you connect artificial intelligence (AI) tools to your clinical practice. We will develop working examples using a cloud-based picture archiving and communication system (PACS), from which we retrieve a study, apply an AI tool, and send the result back to the PACS.

Background

Most of the literature surrounding AI in medicine focuses on training AI systems. This challenge is indeed important, and it is hoped that this series of articles, as well as the many other articles in the literature, have demonstrated that there can be subtle challenges to assuring that the AI tool will perform correctly in the real world.

A part of the problem that has received very little attention is the implementation of AI into clinical practice and that is the focus of this article. In particular, we will connect to a cloud-based PACS (*https://www.dicomserver.co.uk/*), check to make sure the connection is working, query that PACS for a specific study, retrieve that study, apply an AI tool to that study, and send the result back to the PACS. We note that this process still involves a human selecting the study (and series) of interest, and in the real world, using AI to automatically do this step is critical to efficiency and reliability. This article will not cover the issues around access, security, patient privacy, and potential disruption to production systems associated with connecting a research tool to a production PACS. In case the reader wants to execute this code in his or her own institution, we strongly recommend using a research PACS instead to avoid potential negative impact on the production clinical PACS environment.

Description

Please begin by loading the notebook we have created for this at: *https://github.com/RSNA/MagiciansCorner/blob/master/MC8_HowToConnectToPACS.ipynb*. You can do this either by clicking on that link and then the 'Open In Colab' button, or by starting colab.research.google.com and opening from github, RSNA, and MC8_HowToConnectToPACS.ipynb. If the reader is unfamiliar with the Colab environment, we recommend reading the topic "Software Setup" in the first article of the Magician's Corner series *(https://pubs.rsna.org/doi/10.1148/ryai.2019190072)* (1). Because we are connecting to the cloud, it is necessary that you have access to the internet not only to download the code but also while running it.

Once you have opened the notebook in Colab (or your preferred environment), please run cell 1 to install the libraries we need and run cell 2 to import them. Note that there are some new libraries that we will be using. The *pynetdicom* library provides methods to communicate using Digital Imaging and Communications in Medicine (DICOM) (2–4), and we will describe those specific functions in just a bit. We also use the *SimpleITK* library, which provides the ability to load DICOM images and arrange them in a format suitable for doing the AI work. It also provides access to the powerful image processing library called ITK, but we won't describe that further here.

Cell 3 defines the information needed to connect to the PACS. In our case, we are using a public, web-accessible PACS that is located in the United Kingdom and generously provided by DICOM Server. The network address is set on line 5 of the cell. Network communications also have a port number that ranges from 0 to 65535; certain values or ranges of numbers have specific meaning. If you wish to connect to your institution's PACS, it will be necessary to get the local address from your PACS administrator. Perhaps the biggest challenge you may face is being permitted to connect your tool to your clinical PACS, because of concerns either about security (performance, privacy, and data integrity) or about the information you may be putting into the clinical record. Such policy issues are covered elsewhere (5,6). In this article, we hope to provide some technical pieces to enable you to at least consider the policy challenges. A research PACS may be more accessible and may be a better target for your work. Please run cell 3. Note that once you have successfully run cell 3 and connected to the PACS, you cannot run it again and create a new connection: you already have one and the PACS will likely reject the request for an additional connection.

Cell 4 takes the information from cell 3 and checks to see if we can actually connect to the PACS. DICOM provides a function for this called "C-ECHO" which simply sends a "hello" message to the PACS, and the PACS

Summary

In this article, authors show how to use Digital Imaging and Communications in Medicine (DICOM) Query and Retrieve functions to pull a study from a cloud or public picture archiving and communication system (PACS), run an artificial intelligence (AI) algorithm on those images, and store the results back to another (cloud) PACS. This is a practical example of how to both get images needed by an AI tool for inference, and how to store the results back using DICOM methods.

Key Points

- Digital Imaging and Communications in Medicine (DICOM) is the standard way to communicate with clinical imaging systems.
- The technical parts of DICOM Query/Retrieve and Send are approachable but require attention to the details of DICOM protocols.
- One must also be able to convert from DICOM to the file format required by the AI tool being used.
- In addition to the technical aspects of data transfer, the policy issues of public picture archiving and communication system access must be considered.

should respond with its “hello.” Of course, it doesn’t actually use the word hello, but instead we send the specific context under which we want to communicate which is the DICOM context for this hello (line 8), and then we set up an “association” which is the attempt to connect. If the association is accepted by the PACS, it should respond to that context message with its C-ECHO response (0 × 0000 in our case which means success), which is printed below cell 4. Please run cell 4 and confirm that you also get “0 × 0000” as the C-ECHO response. If you don’t, it either means you don’t have a good internet connection, or that the PACS you configured is down or otherwise won’t respond to you.

In cell 5, we define the study that we wish to retrieve. The first step with DICOM is that we query the PACS to find out what it has that matches our search criteria. In our case, we want to retrieve a study, and we know the accession number, so we don’t care what the PatientID or StudyInstanceUID values are. The PatientID is probably familiar to all—it is the identifier that a hospital or clinic assigns to a patient. The StudyInstanceUID is a unique string of characters (UID stands for unique identifier) and should be a globally unique value. It is usually a series of numbers with periods (‘.’) interspersed. We send the asterisk character (‘*’) as a wildcard in both PatientID and StudyInstanceUID because this is the way to indicate to pynetdicom we want that information back. Please run cell 5, and you should then see some of the basic information about the requested study. Note that this does not transfer any data; it just gives us information that the PACS has about that study.

In cell 6, we get more information about the study that is important to transfer the study. In cell 7, we use this information to request that the information be transferred to our DICOM receiver. Note that DICOM is very particular about how things are done; the references provide more information (1–3).

Please run cells 6 and 7, and then run cell 8 to list the images that have been transferred. The second command ‘pipes’ the output of the ‘ls’ command to the word count (‘wc’) command, which counts the number of lines and words (in our case, entries between the periods). You should have 93 lines. Cell 9 will

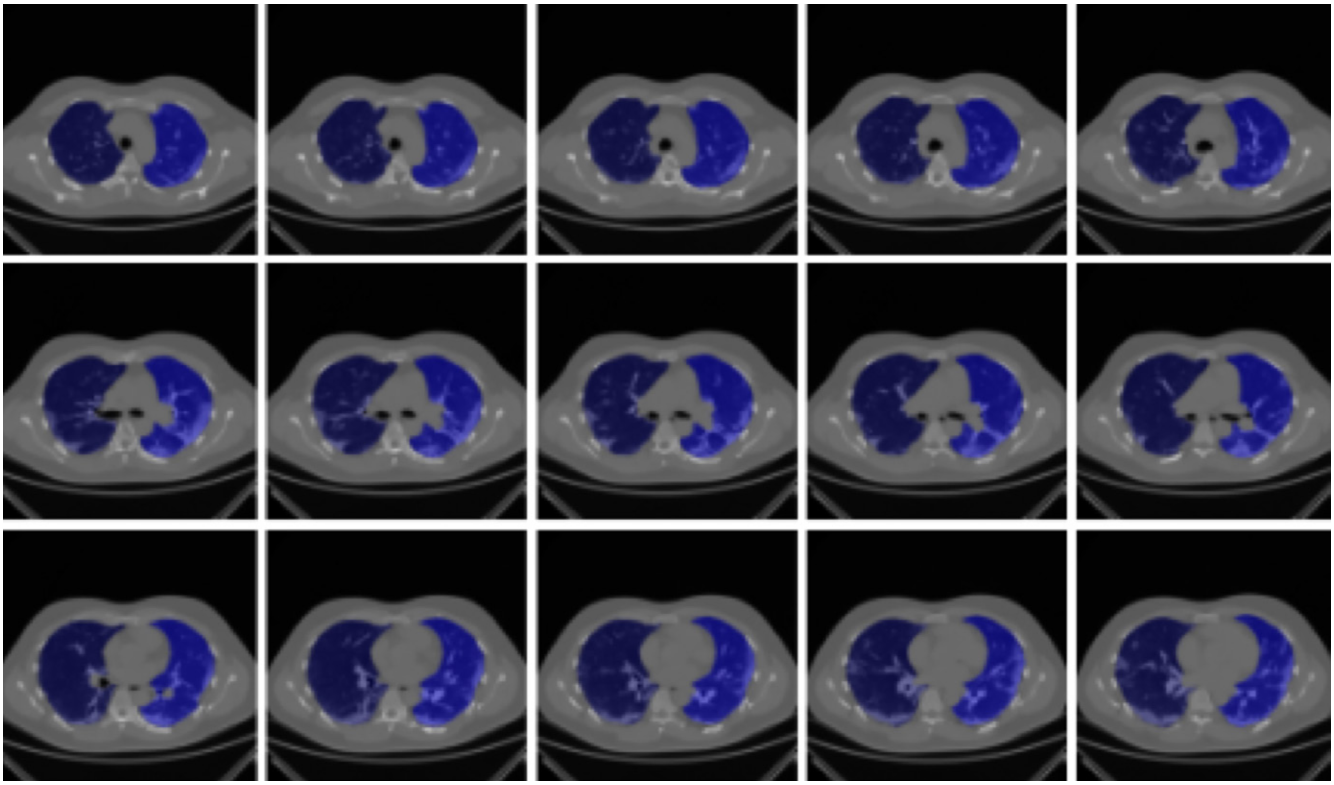
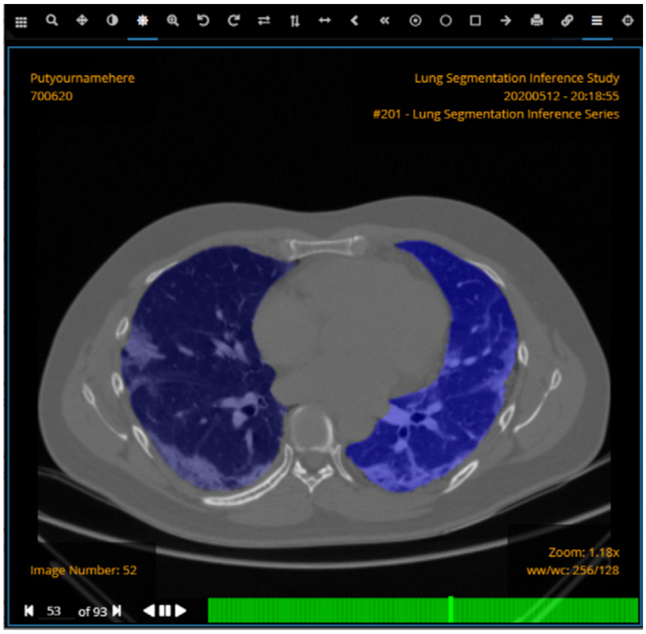


Figure 1: Results of the lung segmentation inference for a few of the images. This displays the CT images, with the inferred lung segmentation in blue.

display the images, and you should see the CT images displayed with soft-tissue window settings. Run cell 9 to see these images. You can adjust the window center and width by editing the value of the variables WC and WW, respectively. Feel free to change them and rerun this cell to see the results, but these changes are for visualization purposes only. They do not affect the input to the deep learning model.

Cell 10 does the actual work of segmentation on the case we just retrieved, using a model that has been previously trained (7). This model is a convolutional neural network called U-Net, commonly used for segmentation tasks. We will use this model to do an "inference," indicating that we are inferring from the prior trained examples what this new unknown example might be. The inference process actually takes relatively little computing power, which you can verify by running it both with and without the graphics processing unit (GPU) (Runtime > Change runtime type > GPU or CPU). Note that you have to run the entire notebook when you change runtime types; you can't just run cell 10 with or without GPU. The exact times you see may differ from what is in this article, because it depends on the specific GPU used. There may be differences in GPU allocations depending on user access and rights. With a GPU, the time required to apply the model to all the CT sections was about 71 seconds, while the time with a CPU was 258 seconds. This time includes writing the data, so the computational benefit is somewhat higher, but points to the fact that for inference, a GPU may not be so important. The second part of this cell displays the CT images with the inferred segmentation of the lungs in blue (Fig 1).



Putyournamehere
700620

Lung Segmentation Inference Study
20200512 - 20:18:55
#201 - Lung Segmentation Inference Series

Image Number: 52

Zoom: 1.18x
ww/wc: 256/128

53 of 93

Figure 2: After the new Digital Imaging and Communications in Medicine (DICOM) images are generated, they are sent to the open picture archiving and communication system (Orthanc). This is a picture of the OSIMIS web viewer showing the DICOM sent by the Colab notebook.

In cell 11, the results of cell 10 are formatted as DICOM objects. That step assures that the DICOM header information (eg, dates and times) is correct. DICOM also wants to have unique identifiers for different information (such as what we have created). Of note, there is a long list of DICOM requirements; we made sure we complied with enough of them to make this example work. You can insert and adjust for your circumstances. Feel free to change the PatientName string "Putyournamehere" to a new unique one so that you can identify *your* inference in the (shared) PACS later. Please run cell 11.

Cells 12 and 13 take the DICOM images that were created in cell 11 and perform a C-MOVE operation (DICOM terminology for sending data), causing the transfer of the images to the PACS. Note that there are likely others reading this article who are also creating images and sending them to this same PACS. For that reason, we create a random identifier in cell 11 used for making different instances, although the content should be the same. If the same ID was used, each user would overwrite the results of a prior user.

After the last cell is a comment cell that shows you how you can access the cloud PACS using an application called Orthanc and a web viewer called Osimis (Fig 2). Login into the PACS with the credentials provided (user 'orthanc' and password 'orthanc') and look for the PatientName you provided above. You should then access the web viewer to see your inference result. The end-to-end process is depicted in Figure 3.

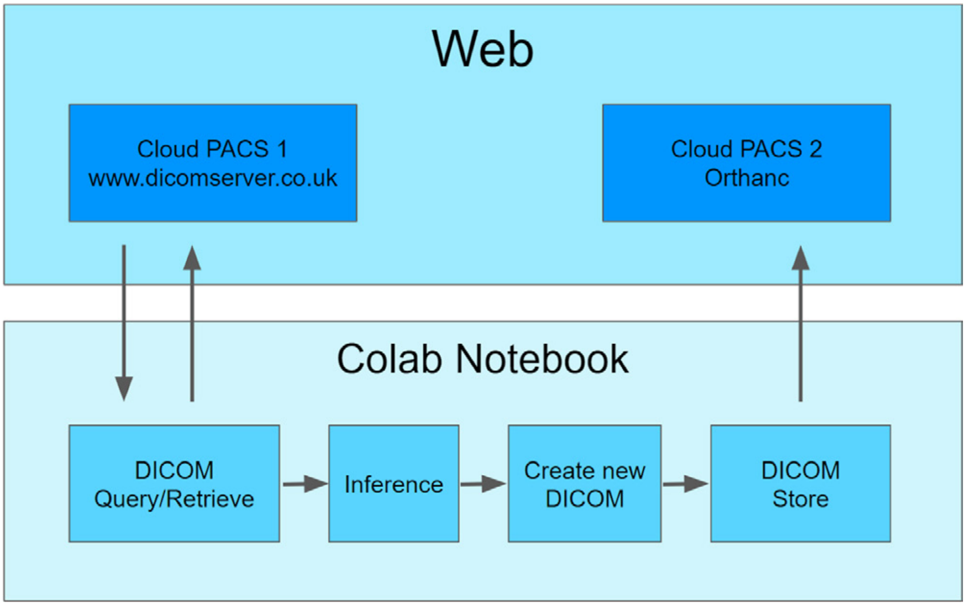


Figure 3: End-to-end process of the toy artificial intelligence model deployment on a cloud picture archiving and communication system.

Conclusion

Integrating AI models in the workflow can be challenging in several aspects, including policy and technical issues. We aimed to provide an example of the technical side of AI model deployment. Although this example lacks many important components such as authentication and auditing or considerations of performance, it provides a functioning example of how to retrieve an examination from a PACS, apply an AI tool to it ('inference'), and send the result back to the PACS for review.

Author contributions: Guarantors of integrity of entire study, B.J.E., F.K.; study concepts/study design or data acquisition or data analysis/interpretation, all authors; manuscript drafting or manuscript revision for important intellectual content, all authors; approval of final version of submitted manuscript, all authors; agrees to ensure any questions related to the work are appropriately resolved, all authors; literature research, B.J.E., F.K.; experimental studies, F.K.; and manuscript editing, B.J.E., F.K.

Disclosures of Conflicts of Interest: **B.J.E.** disclosed no relevant relationships. **F.K.** Activities related to the present article: disclosed no relevant relationships. Activities not related to the present article: consultant for MD.ai; employed by DASA as Head of AI. Other relationships: disclosed no relevant relationships.

References

1. Erickson BJ. Magician's Corner: How to start learning about deep learning. Radiol Artif Intell 2019;1(4):e190072.
2. Bidgood WD Jr, Horii SC. Introduction to the ACR-NEMA DICOM standard. RadioGraphics 1992;12(2):345–355.
3. Horii SC. Primer on computers and information technology. Part four: A nontechnical introduction to DICOM. RadioGraphics 1997;17(5):1297–1309.
4. Mildenberger P, Eichelberg M, Martin E. Introduction to the DICOM standard. Eur Radiol 2002;12(4):920–927.
5. Mongan J, Kohli M. Artificial intelligence and human life: five lessons for radiology from the 737 MAX disasters. Radiol Artif Intell 2020;2(2):e190111.
6. Filice RW, Ratwani RM. The case for user-centered artificial intelligence in radiology. Radiol Artif Intell 2020;2(3):e190095.
7. Hofmanninger J, Prayer F, Pan J, Rohrich S, Prosch H, Langs G. Automatic lung segmentation in routine imaging is a data diversity problem, not a methodology problem. arXiv:2001.11767 [preprint]. https://arxiv.org/abs/2001.11767. Posted 2020. Accessed March 3, 2020.